Refine Search

Search Results -

Terms	Documents
L6 not L4	2

3.13.04

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L7

Refine Search

Recall Text Clear Interrupt

Search History

DATE: Saturday, March 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L7	L6 not L4	2	L7
L6	L5 and L3	3	L6
L5	707/$.ccls.	19385	L5
L4	L3 and ((database$1 or (data near base$1) or table$1) near call)	1	L4
L3	(creat$3 or generat$3 or construct$3 or build$3 or form$3) and (database$1 or (data near base$1) or table$1) and (bridge near map)	47	L3
L2	(creat$3 or generat$3 or construct$3 or build$3 or form$3) same (database$1 or (data near base$1) or table$1) same (bridge near map)	2	L2
DB=USPT; PLUR=YES; OP=OR			
L1	(creat$3 or generat$3 or construct$3 or build$3 or form$3) same (database$1 or (data near base$1) or table$1) same (bridge near map)	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

Refine Search

Search Results -

Terms	Documents
L3 and ((database$1 or (data near base$1) or table$1) near call)	1

1, 3, 3. 04

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L4

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Saturday, March 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR			
L4	L3 and ((database$1 or (data near base$1) or table$1) near call)	1	L4
L3	(creat$3 or generat$3 or construct$3 or build$3 or form$3) and (database$1 or (data near base$1) or table$1) and (bridge near map)	47	L3
L2	(creat$3 or generat$3 or construct$3 or build$3 or form$3) same (database$1 or (data near base$1) or table$1) same (bridge near map)	2	L2
DB=USPT; PLUR=YES; OP=OR			
L1	(creat$3 or generat$3 or construct$3 or build$3 or form$3) same (database$1 or (data near base$1) or table$1) same (bridge near map)	1	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 1 of 1 returned.

☐ 1. Document ID: US 20020188616 A1

Using default format because multiple data bases are involved.

L4: Entry 1 of 1 File: PGPB Dec 12, 2002

PGPUB-DOCUMENT-NUMBER: 20020188616
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020188616 A1

TITLE: Database access bridge system and process

PUBLICATION-DATE: December 12, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Chinnici, Roberto R.	Sunnyvale	CA	US	
Rodham, Ken	Gilroy	CA	US	

US-CL-CURRENT: 707/102

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw De

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Terms	Documents
L3 and ((database$1 or (data near base$1) or table$1) near call)	1

Display Format: - Change Format

Previous Page Next Page Go to Doc#

h e b b g e e e f e h e ef b e

Refine Search

Search Results -

Terms	Documents
L14 and (L1 or L2)	18

3.13.04

Database:
- US Pre-Grant Publication Full-Text Database
- US Patents Full-Text Database
- US OCR Full-Text Database
- EPO Abstracts Database
- JPO Abstracts Database
- Derwent World Patents Index
- IBM Technical Disclosure Bulletins

Search: L15

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Saturday, March 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT; PLUR=YES; OP=OR		
L15	L14 and (L1 or L2)	18	L15
L14	L13 not L12	34	L14
L13	L11 and (EJB or (enterprise near java near bean))	43	L13
L12	L11 and (language near programming near call)	16	L12
L11	707/$.ccls.	12381	L11
L10	L4 and (EJB or (enterprise near java near bean))	27	L10
L9	L8 and (EJB or (enterprise near java near bean))	48	L9
L8	(L1 or L2) and call	2656	L8
L7	(L6 or L5) and EJB	27	L7
L6	L4 and mapp$3	34	L6
L5	L3 and mapp$3	6	L5
L4	(L1 or L2) and (language near programming near call)	36	L4
L3	(L1 or L2) and (programming near call)	10	L3
L2	SQL or (structured near query near logic)	4368	L2

h e b b cg b e e ch

L1	SQL or (structured near query near language)	4414	L1

END OF SEARCH HISTORY

h e b b cg b e e ch

WEST

Generate Collection | Print

Search Results - Record(s) 1 through 9 of 9 returned.

☐ 1. Document ID: US 6442748 B1

L15: Entry 1 of 9 File: USPT Aug 27, 2002

US-PAT-NO: 6442748
DOCUMENT-IDENTIFIER: US 6442748 B1

TITLE: System, method and article of manufacture for a persistent state and persistent object separator in an information services patterns environment

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 2. Document ID: US 6438594 B1

L15: Entry 2 of 9 File: USPT Aug 20, 2002

US-PAT-NO: 6438594
DOCUMENT-IDENTIFIER: US 6438594 B1

TITLE: Delivering service to a client via a locally addressable interface

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 3. Document ID: US 6266716 B1

L15: Entry 3 of 9 File: USPT Jul 24, 2001

US-PAT-NO: 6266716
DOCUMENT-IDENTIFIER: US 6266716 B1

TITLE: Method and system for controlling data acquisition over an information bus

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 4. Document ID: US 6226788 B1

L15: Entry 4 of 9 File: USPT May 1, 2001

US-PAT-NO: 6226788
DOCUMENT-IDENTIFIER: US 6226788 B1

TITLE: Extensible network management system

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

☐ 5. Document ID: US 6205465 B1

L15: Entry 5 of 9 File: USPT Mar 20, 2001

US-PAT-NO: 6205465
DOCUMENT-IDENTIFIER: US 6205465 B1

TITLE: Component extensible parallel execution of multiple threads assembled from program components specified with partial inter-component sequence information

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 6. Document ID: US 5991751 A

L15: Entry 6 of 9 File: USPT Nov 23, 1999

US-PAT-NO: 5991751
DOCUMENT-IDENTIFIER: US 5991751 A

TITLE: System, method, and computer program product for patent-centric and group-oriented data processing

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 7. Document ID: US 5428782 A

L15: Entry 7 of 9 File: USPT Jun 27, 1995

US-PAT-NO: 5428782
DOCUMENT-IDENTIFIER: US 5428782 A

TITLE: Portable and dynamic distributed applications architecture

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 8. Document ID: US 5257366 A

L15: Entry 8 of 9 File: USPT Oct 26, 1993

US-PAT-NO: 5257366
DOCUMENT-IDENTIFIER: US 5257366 A

TITLE: Query language execution on heterogeneous database servers using a bind-file bridge between application and database languages

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | KWIC | Draw Desc | Image

---

☐ 9. Document ID: US 5179660 A

L15: Entry 9 of 9 File: USPT Jan 12, 1993

US-PAT-NO: 5179660
DOCUMENT-IDENTIFIER: US 5179660 A

TITLE: System for reducing communications overhead in distributed database transactions by serializing in order related requests into single transmission message and receiving transmission response

Generate Collection | Print

Terms	Documents
L14 and L11	9

Display Format: TI | Change Format

Previous Page Next Page

L11: Entry 4 of 5 File: USPT Mar 30, 1999

DOCUMENT-IDENTIFIER: US 5890160 A
TITLE: Object representation of relational database cells having nontraditional large object datatypes

Brief Summary Text (4):
Relational database products, which are used in computer systems, now support nontraditional column datatypes such as audio and video. As used here, the term "computer systems" encompasses the widest possible meaning and includes, but is not limited to, standalone processors, networked processors, mainframe processors, processors in a client/server relationship, and embedded processors. When object oriented applications access rows of relational tables, the data value of the cell is retrieved into an object. The object makes available the value of the nontraditional datatype, but the manipulation of the value via behaviors or methods of the object is not accommodated. Such manipulation presently is possible only through calls to the database server. Thus, object oriented applications programs cannot access and manipulate nontraditional type data values from relational tables according to the object oriented programming paradigm.

Detailed Description Text (5):
In an environment having a database management system, applications programs communicate with an automated database manager. The database manager may be referred to as a database server. In particular, the applications programs may send messages to the database server in a predefined format. Such formatted messages may be referred to as database calls. A database call invokes one or more corresponding functions of the database management system, usually with respect to a particular database. A database management system provides applications programs with a variety of callable functions.

Detailed Description Text (14):
The function calls that an applications program may make to the database server have a somewhat standardized structure that is tailored to the relational model. This structure for RDBMS function calls is generally referred to as the Structured Query Language (SQL).

Detailed Description Text (25):
As mentioned above, applications programs access the data of relational tables by making calls to a database server. Used in this sense, the term "applications programs" may refer to several separate programs, only one program, a module of a program, or even a particular task of a module.

Detailed Description Text (45):
An OO applications program may access the data stored in a relational table by making function calls to the database server of the RDBMS in SQL (see FIG. 2). For example, a class might be defined to have behaviors B that: generate appropriate SQL statements; package the statements and forward them to the database server; receive the results; process the results; and so on. When such a program is executed, client objects C.sub.-- Obj of the foregoing class would be constructed as necessary, and their behaviors B invoked in accordance with the particular task.

Detailed Description Text (47):
This approach, however, is not desirable. In particular, the applications programmer must have an intimate knowledge of the RDBMS and its SQL function calls. The applications programs and the RDBMS become tightly coupled under this basic

approach. A change to the RDBMS, therefore, often requires extensive changes in all of the tightly-coupled applications programs.

Detailed Description Text (48):
Another disadvantage of this approach is that applications programmers must depart from the object model with respect to accessing and manipulating data from the relational tables. Applications programmers must instead use and apply the relational model. In particular, applications programmers must fully understand SQL, must have detailed knowledge of the database schema, and must "switch" their thinking from the object model to the relational model.

Detailed Description Text (50):
An alternative approach is to provide an OO interface between the RDBMS and applications programs (see FIG. 3). That is, instead of making function calls directly to the database server of the RDBMS, a client object C.sub.-- Obj may pass an appropriate message to an intermediate OO access facility (OOAO, for object oriented access object) which is responsible for direct communication with the database server.

Detailed Description Text (55):
Under this approach, client objects may be simplified because the OOAO has the responsibility for intimate knowledge of the database schema, of the precise syntax of SQL calls, and of the method of communicating with the database server. Applications programmers are thus freed from such responsibility, and can operate more completely under the object model. One important way in which the OOAO allows more complete operation under the object model is that, since the OOAO is itself an object, client objects communicate with it in the OO manner of passing messages to invoke member functions B.

Detailed Description Text (72):
Some RDBMS products provide server-based functions that manipulate data of a nontraditional datatype. To invoke such server-based functions, an appropriate SQL statement must be provided. For example, an SQL statement that might provide the desired rotation of an image data value could appear as follows:

Detailed Description Text (75):
According to this method, the image that originally was retrieved and found to be in need of rotation is not actually rotated. What happens with the use of server-based functions is that the image is re-retrieved from the table in the server, is rotated by the server, and is then provided in response to the SQL statement. The image manipulation occurs at the database server, and is performed by the RDBMS.

Detailed Description Text (104):
In the presently preferred embodiment of the invention, managing the storage and retrieval of LOBs and the generation of LOB locators is a function of the RDBMS. Since the RDBMS is responsible for these functions, SQL statements must normally be provided to the RDBMS so as selectably to retrieve a LOB or a LOB locator.

Detailed Description Text (115):
Using the indirect LAM, a LOB locator generated on the basis of the location of the LOB in the first table would have been provided by the RDBMS from the server to the client's applications program, and then the LOB locator subsequently could have been used by the applications program to cause the RDBMS to store the actual LOB in the desired location in the second table. Since the LOB locator has a length that is negligible in comparison to the length of a LOB, virtually no network impact would have been realized.

Detailed Description Text (129):
Therefore, even though the OO applications programmer may wish to have LOB's be kept at the server, and may use LOB locators, the programmer may remain squarely within the OO paradigm. The applications programmer need not use SQL to cause generation of LOB locators, because this is taken care of by the member functions of the LOB EXOB subclass.